

Detecting and Reporting Object-Relational Mapping Problems: An Industrial Report



Marcos Nazário



Eduardo Guerra



Rodrigo Bonifácio



Gustavo Pinto



**Application
Code**

ORM

Database



```
@Entity
public class Color {

    private String code;

    private String name;

    private Boolean enable;
}
```

```
@Entity
```

```
@Table(schema = "public", name = "COLOR")
```

```
public class Color {
```

```
    private String code;
```

```
    private String name;
```

```
    private Boolean enable;
```

```
}
```



```
@Entity
@Table(schema = "public", name = "COLOR")
public class Color {

    @Column
    private String code;

    @Column
    private String name;

    @Column
    private Boolean enable;
}
```

```
@Entity
```

```
@Table(schema = "public", name = "COLOR")
```

```
public class Color {
```

```
    @Column(name = "CODE", nullable = false)
```

```
    private String code;
```

```
    @Column(name = "NAME", length = 80, nullable = false)
```

```
    private String name;
```

```
    @Column(name = "ENABLE", nullable = false)
```

```
    private Boolean enable;
```

```
}
```

```
@Entity
@Table(schema = "public", name = "COLOR")
public class Color {

    @Column(name = "CODE", nullable = false)
    private String code;

    @Column(name = "NAME", length = 80, nullable = false)
    private String name;

    @Column(name = "ENABLE", nullable = false)
    private Boolean enable;
}
```



```
@Entity
@Table(schema = "public", name = "COLOR")
public class Color {

    @Column(name = "CODE", nullable = false)
    private String code;

    @Column(name = "NAME", length = 80, nullable = false)
    private String name;

    @Column(name = "ENABLE", nullable = false)
    private Boolean enable;
}
```

Problem with id column and JPA

Asked 8 years, 4 months ago Active 6 years, 6 months ago Viewed 7k times



I'm trying JPA with a very simple class for the Play! framework and I'm having some problems with the id column.

4

My sql database has only two columns:



```
CREATE TABLE IF NOT EXISTS `auto` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```

And my model is:

```
@Entity  
@Table(name = "auto")  
public class Auto extends Model{  
  
    @Column(insertable = false, updatable = false)  
    public int id;  
  
    public String name;  
  
    public Auto(String name){  
        this.name = name;  
    }  
}
```

Everything works fine without this part:

```
@Entity  
@Table(sche  
public clas  
  
@Column  
private  
  
@Column  
private  
  
@Column  
private  
  
}
```

```
@Entity
@Table(schema = "public", name = "COLOR")
public class Color {

    @Id
    @Column(name = "CODE", nullable = false)
    private String code;

    @Column(name = "NAME", length = 80, nullable = false)
    private String name;

    @Column(name = "ENABLE", nullable = false)
    private Boolean enable;
}
```



```
@Entity
@Table(schema = "public", name = "COLOR")
public class Color {

    @Id
    @Column(name = "CODE", nullable = false)
    private String code;

    @Column(name = "NAME", length = 80, nullable = false)
    private String name;

    @Column(name = "ENABLE", nullable = false)
    private Boolean enable;
}
```


Cannot be cast to java.io.Serializable

Asked 8 years, 8 months ago Active 1 year, 4 months ago Viewed 30k times



16



3

I am currently using criteria to retrieve the details of a user, but when trying to query the details object with the right user, I get a ClassCastException.

My Criteria Code;

```
Criteria criteria = sess.createCriteria(UserDetails.class)
criteria.add(Restrictions.eq("user.id", user.id));
```

I also tried using;

```
Criteria criteria = sess.createCriteria(UserDetails.class)

Criteria subCriteria = criteria.createCriteria("user");
subCriteria.add(Restrictions.eq("id", user.id));
```

Both give me the ClassCastException. I know I can easily solve it by letting the User implement Serializable, but is there any other solution?

[java](#) [hibernate](#) [criteria](#)

[share](#) [improve this question](#)

asked Dec 24 '10 at 8:45



[Thys](#)

5,653 ● 25 ● 82 ● 130

Why you do not want to implement Serializable? – [Juha Syrjälä](#) Dec 24 '10 at 9:40

2 I have the feeling that there is a better way to solve it – [Thys](#) Dec 24 '10 at 9:42

Have you checked this link: stackoverflow.com/questions/2020904/... – [craftsman](#) Dec 24 '10 at 11:30

@Entity

@Table (schema = "public")

public class

@Id

@Column

private

@Column

private

@Column

private

}



@gustavopinto

```
@Entity
@Table(schema = "public", name = "COLOR")
public class Color implements Serializable {

    @Id
    @Column(name = "CODE", nullable = false)
    private String code;

    @Column(name = "NAME", length = 80, nullable = false)
    private String name;

    @Column(name = "ENABLE", nullable = false)
    private Boolean enable;
}
```

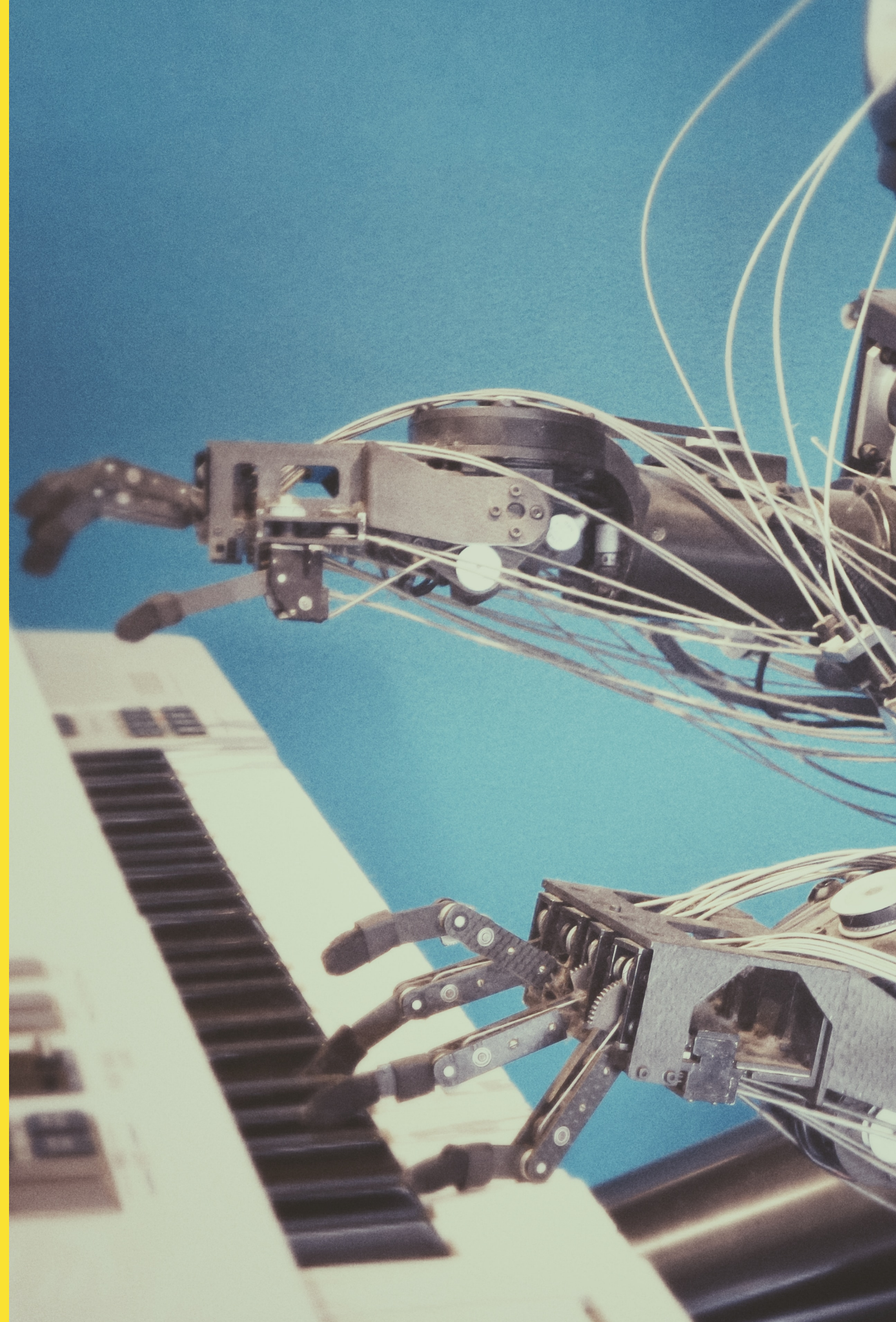


We catalogued
12
ORM-related
problems



@gustavopinto

**We implemented
this catalog in a
static analysis
tool**



**We implemented
this catalog in a
static analysis
tool**

Finds the problems

Suggests the fix

**We implemented
this catalog in a
static analysis
tool**

**We are using this
tool for over
5 years
now**

Finds the problems

Suggests the fix



**Could
others take
advantage
of our tool?**



**We conducted a
developer
experience (DX)
study with 13
developers**

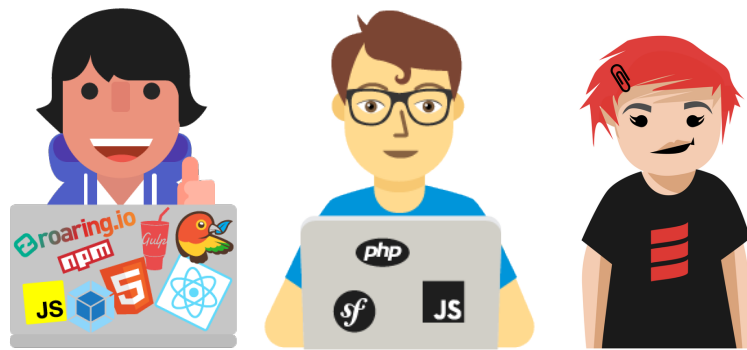


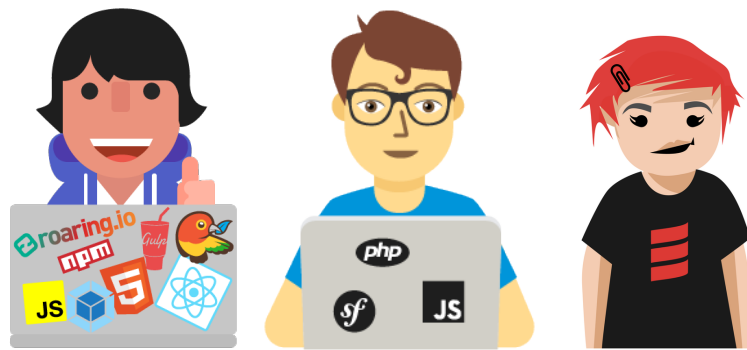
We conducted a
developer
experience (DX)
study with 13
developers

3 students,
10 from industry

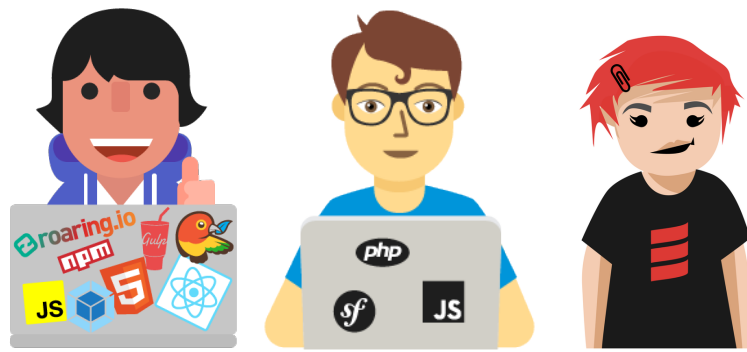
6 with **7+ years** with
ORM experience

2 with **no previous**
ORM experience

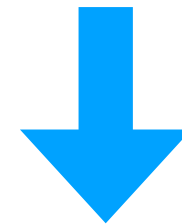
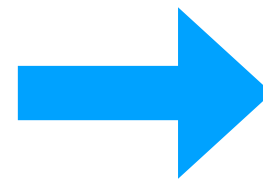


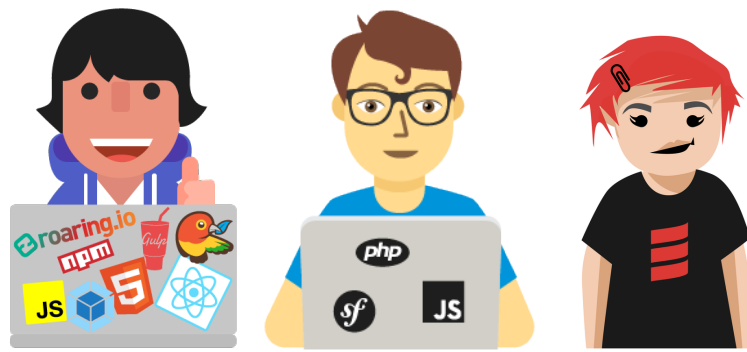


6 ORM tasks

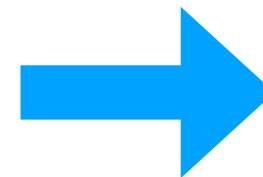
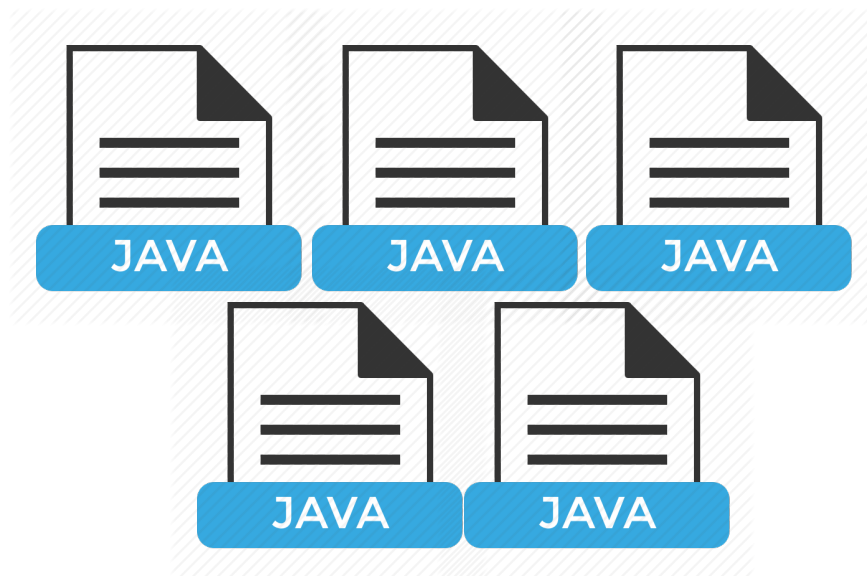
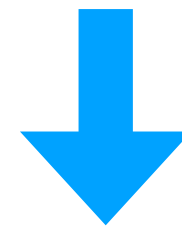
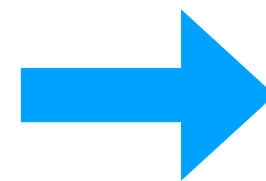


6 ORM tasks





6 ORM tasks





**All participants completed the
experiment**



eases ORM modeling

**“ Every day I do object-relational
mapping, and this tool
could help me to remember
what is missing ”**

eases ORM modeling

the tool is precise

forces best practices





Little docs

**“ I found it quite difficult to
find help in the
documentation. There is
nothing on the web ”**



Little docs

**Custom annotations
were not intuitive**

**The tool poses many
constraints**

@Entity
@Table(sche
public clas

@Id
@Column
private

@Column
private

@Column
private

}

Cannot be cast to java.io.Serializable

Asked 8 years, 8 months ago · Active 1 year, 4 months ago · Viewed 30k times

▲

16

▼

★

3

I am currently using criteria to retrieve the details of a user, but when trying to query the details object with the right user, I get a ClassCastException.

My Criteria Code;

```
Criteria criteria = sess.createCriteria(UserDetails.class)
criteria.add(Restrictions.eq("user.id", user.id));
```

I also tried using;

```
Criteria criteria = sess.createCriteria(UserDetails.class)
Criteria subCriteria = criteria.createCriteria("user");
subCriteria.add(Restrictions.eq("id", user.id));
```

Both give me the ClassCastException. I know I can easily solve it by letting the User implement Serializable, but is there any other solution?

java · hibernate · criteria

share · improve this question

asked Dec 24 '10 at 8:45

Thys

5,693 · 25 · 82 · 130

Why you do not want to implement Serializable? – Juha Syrjäla

Dec 24 '10 at 9:40

2 I have the feeling that there is a better way to solve it – Thys

Dec 24 '10 at 9:42

Have you checked this link: stackoverflow.com/questions/2020904/... – craftsman

Dec 24 '10 at 11:30

```
@Entity
@Table(scher
public class

@Id
@Columnn
private

@Columnn
private

@Columnn
private

}
```

Cannot be cast to java.io.Serializable

Asked 8 years, 8 months ago · Active 1 year, 4 months ago · Viewed 30k times

16

I am currently using criteria to retrieve the details of a user, but when trying to query the details object with the right user, I get a ClassCastException.

My Criteria Code;

```
Criteria criteria = sess.createCriteria(UserDetails.class)
criteria.add(Restrictions.eq("user.id", user.id));
```

I also tried using;

```
Criteria criteria = sess.createCriteria(UserDetails.class)
Criteria subCriteria = criteria.createCriteria("user");
subCriteria.add(Restrictions.eq("id", user.id));
```

Both give me the ClassCastException. I know I can easily solve it by letting the User implement Serializable, but is there any other solution?

java hibernate criteria

share improve this question

asked Dec 24 '10 at 8:45

Thys 5,693 25 82 130

Why you do not want to implement Serializable? – Juha Syrjäla Dec 24 '10 at 9:40

2 I have the feeling that there is a better way to solve it – Thys Dec 24 '10 at 9:42

Have you checked this link: stackoverflow.com/questions/2020904/ – craftsman Dec 24 '10 at 11:30

@gustavopinto

We implemented this catalog in a static analysis tool

We are using this tool for over 5 years now

@gustavopinto


```
@Entity
@Table(schema = "users")
public class User {

    @Id
    @Column(name = "id")
    private Long id;

    @Column(name = "username")
    private String username;

    @Column(name = "password")
    private String password;
}
```

Cannot be cast to java.io.Serializable

Asked 8 years, 8 months ago · Active 1 year, 4 months ago · Viewed 30k times

16

I am currently using criteria to retrieve the details of a user, but when trying to query the details object with the right user, I get a ClassCastException.

My Criteria Code;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
criteria.add(Restrictions.eq("user.id", user.id));
```

I also tried using;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
Criteria subCriteria = criteria.createCriteria("user");
subCriteria.add(Restrictions.eq("id", user.id));
```

Both give me the ClassCastException. I know I can easily solve it by letting the User implement Serializable, but is there any other solution?

java hibernate criteria

share improve this question

asked Dec 24 '10 at 8:45

Thijs 5,693 ● 25 ● 82 ● 130

Why you do not want to implement Serializable? – Juha Syrjäla Dec 24 '10 at 9:40

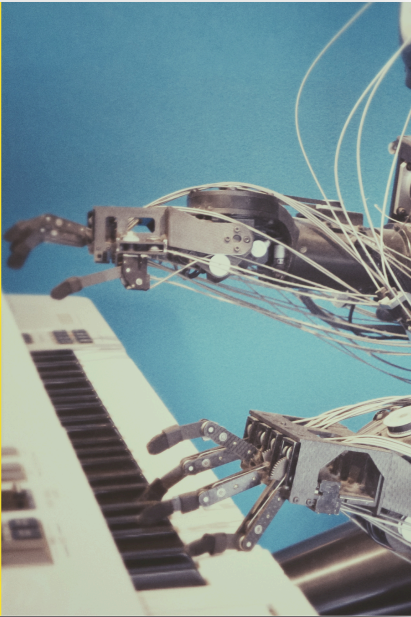
2 I have the feeling that there is a better way to solve it – Thijs Dec 24 '10 at 9:42

Have you checked this link: stackoverflow.com/questions/2020904/ – craftsman Dec 24 '10 at 11:30

@gustavopinto


We implemented this catalog in a static analysis tool

We are using this tool for over 5 years now



@gustavopinto

We conducted a developer experience (DX) study with 13 developers



10 from industry, 3 students

6 with 7+ years with ORM experience

2 with no previous ORM experience

@gustavopinto

```
@Entity
@Table(schema = "users")
public class User {

    @Id
    @Column(name = "id")
    private Long id;

    @Column(name = "username")
    private String username;

    @Column(name = "password")
    private String password;
}
```

Cannot be cast to java.io.Serializable

Asked 8 years, 8 months ago · Active 1 year, 4 months ago · Viewed 30k times

16

I am currently using criteria to retrieve the details of a user, but when trying to query the details object with the right user, I get a ClassCastException.

My Criteria Code;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
criteria.add(Restrictions.eq("user.id", user.id));
```

I also tried using;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
Criteria subCriteria = criteria.createCriteria("user");
subCriteria.add(Restrictions.eq("id", user.id));
```

Both give me the ClassCastException. I know I can easily solve it by letting the User implement Serializable, but is there any other solution?

java hibernate criteria

share improve this question

asked Dec 24 '10 at 8:45

Thys 5,693 25 82 130

Why you do not want to implement Serializable? – Juha Syrjä Dec 24 '10 at 9:40

2 I have the feeling that there is a better way to solve it – Thys Dec 24 '10 at 9:42

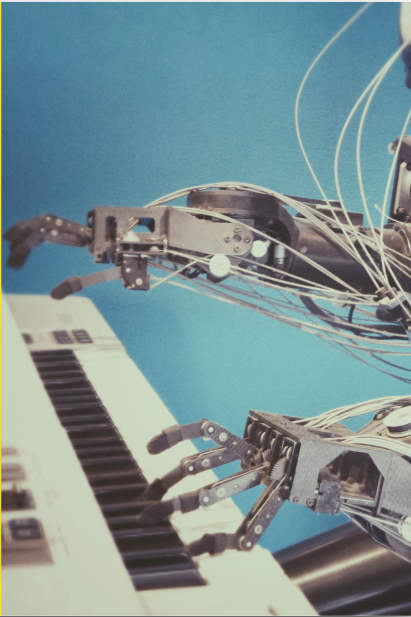
Have you checked this link: stackoverflow.com/questions/2020904/ – craftsman Dec 24 '10 at 11:30

@gustavopinto

We implemented this catalog in a static analysis tool


We are using this tool for over 5 years now

@gustavopinto



We conducted a developer experience (DX) study with 13 developers

@gustavopinto



- 10 from industry, 3 students
- 6 with 7+ years with ORM experience
- 2 with no previous ORM experience

eases ORM modeling

the tool is precise

forces best practices



@gustavopinto


```
@Entity
@Table(schema = "public")
public class User {

    @Id
    @Column(name = "id")
    private Long id;

    @Column(name = "username")
    private String username;

    @Column(name = "password")
    private String password;
}
```

Cannot be cast to java.io.Serializable

Asked 8 years, 8 months ago · Active 1 year, 4 months ago · Viewed 30k times

16

I am currently using criteria to retrieve the details of a user, but when trying to query the details object with the right user, I get a ClassCastException.

My Criteria Code;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
criteria.add(Restrictions.eq("user.id", user.id));
```

I also tried using;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
Criteria subCriteria = criteria.createCriteria("user");
subCriteria.add(Restrictions.eq("id", user.id));
```

Both give me the ClassCastException. I know I can easily solve it by letting the User implement Serializable, but is there any other solution?

java hibernate criteria

share improve this question

asked Dec 24 '10 at 8:45

Thys 5,693 25 82 130

Why you do not want to implement Serializable? – Juha Syrjä Dec 24 '10 at 9:40

2 I have the feeling that there is a better way to solve it – Thys Dec 24 '10 at 9:42

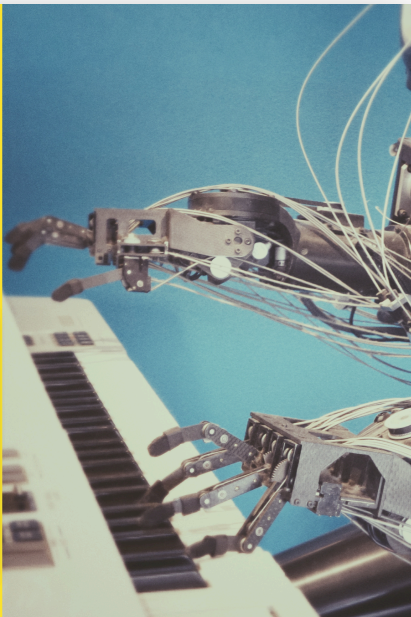
Have you checked this link: stackoverflow.com/questions/2020904/ – craftsman Dec 24 '10 at 11:30

@gustavopinto

We implemented this catalog in a static analysis tool

We are using this tool for over 5 years now

@gustavopinto




We conducted a developer experience (DX) study with 13 developers

10 from industry, 3 students

6 with 7+ years with ORM experience

2 with no previous ORM experience

@gustavopinto



eases ORM modeling

the tool is precise

forces best practices

@gustavopinto

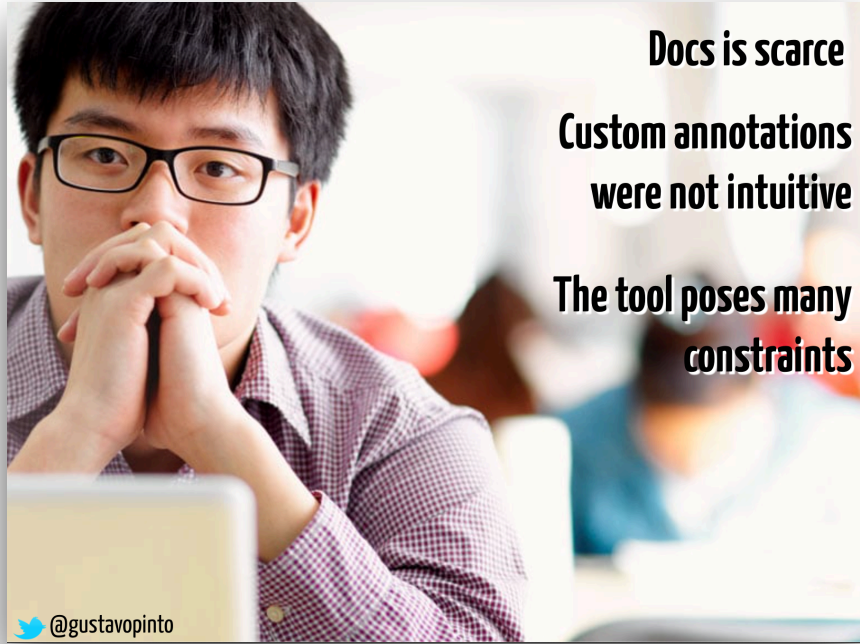


Docs is scarce

Custom annotations were not intuitive

The tool poses many constraints

@gustavopinto




```
@Entity
@Table(schema = "public")
public class User {

    @Id
    @Column(name = "id")
    private Long id;

    @Column(name = "name")
    private String name;

    @Column(name = "email")
    private String email;
}
```

Cannot be cast to java.io.Serializable

Asked 8 years, 8 months ago · Active 1 year, 4 months ago · Viewed 30k times

16

I am currently using criteria to retrieve the details of a user, but when trying to query the details object with the right user, I get a ClassCastException.

My Criteria Code;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
criteria.add(Restrictions.eq("user.id", user.id));
```

I also tried using;

```
Criteria criteria = sess.createCriteria(UserDetails.class);
Criteria subCriteria = criteria.createCriteria("user");
subCriteria.add(Restrictions.eq("id", user.id));
```

Both give me the ClassCastException. I know I can easily solve it by letting the User implement Serializable, but is there any other solution?

java hibernate criteria

share improve this question

asked Dec 24 '10 at 8:45

Thys 5,693 25 82 130

Why you do not want to implement Serializable? – Juha Syrjä Dec 24 '10 at 9:40

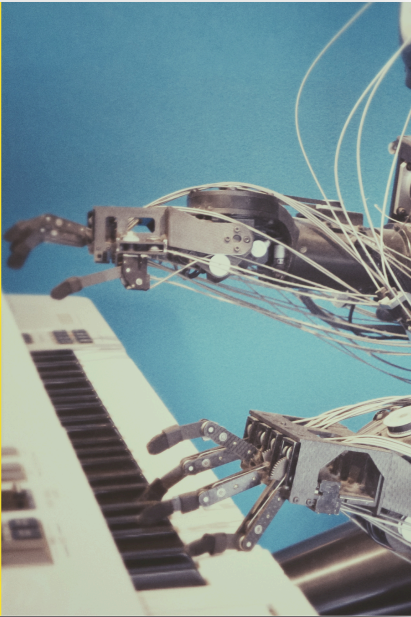
2 I have the feeling that there is a better way to solve it – Thys Dec 24 '10 at 9:42

Have you checked this link: stackoverflow.com/questions/2020904/... – craftsman Dec 24 '10 at 11:30

@gustavopinto

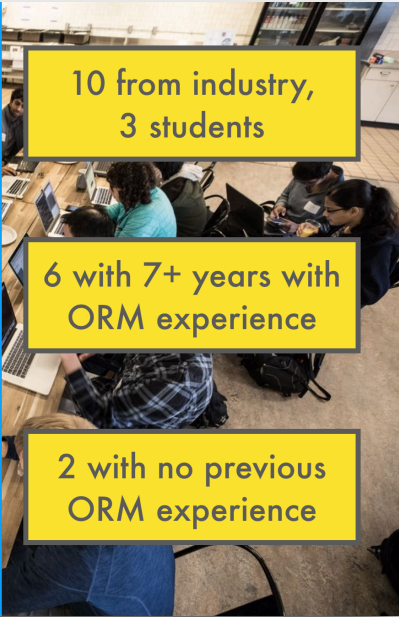
We implemented this catalog in a static analysis tool

We are using this tool for over 5 years now



@gustavopinto

We conducted a developer experience (DX) study with 13 developers



10 from industry, 3 students

6 with 7+ years with ORM experience

2 with no previous ORM experience

@gustavopinto

eases ORM modeling

the tool is precise

forces best practices

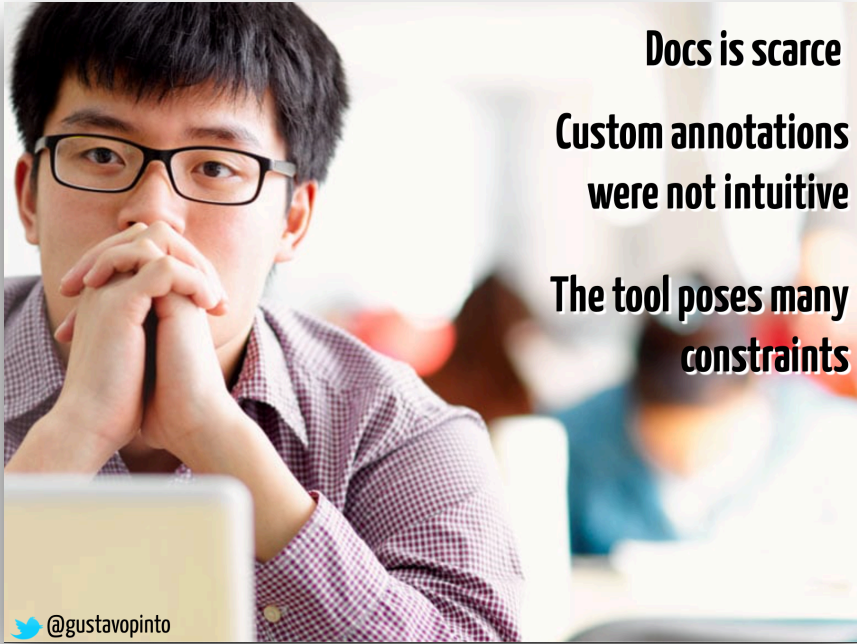


@gustavopinto

Docs is scarce

Custom annotations were not intuitive

The tool poses many constraints



@gustavopinto

Email

Twitter

mail@gustavopinto.org

Web site